

## THE CLAIMS

What is claimed is:

- 1 1. A machine readable medium having stored thereon a plurality of control signals and  
2 when accessed by a processor, causing said processor to:  
3 access a first set of bits as representing an instruction of a first plurality of  
4 instructions, said first plurality of instructions including a packed data instruction, a  
5 scalar floating point instruction, and a transition instruction to be executed between  
6 said packed data instruction and said scalar floating point instruction;  
7 if said first set of bits represents the packed data instruction then generate a first  
8 set of control signals to cause said processor to execute the packed data instruction  
9 on packed data contents of a storage representing a programmer visible register file,  
10 wherein said storage representing the programmer visible register file is operated as a  
11 flat register file while executing said packed data instruction;  
12 if said first set of bits represents the scalar floating point instruction then generate  
13 the first set of control signals to cause said processor to execute the scalar floating  
14 point instruction on floating point contents of the storage representing the  
15 programmer visible register file, wherein said programmer visible register file is  
16 operated as a stack while executing said scalar floating point instruction; and  
17 if said first set of bits represents the transition instruction then generate the first  
18 set of control signals to cause said processor to alter a tag data to indicate that the  
19 stack of the programmer visible register file is empty responsive to executing said  
20 transition instruction between execution of said packed data instruction and said  
21 scalar floating point instruction.

- 1    2. The machine readable medium Claim 1, when accessed by said processor, further  
2        causing the processor to:  
3            generate the first set of control signals to cause said processor to alter a sign and  
4        exponent data to indicate that a floating point contents of the programmer visible  
5        register file represent not-a-number (NaN) if said first set of bits represents the  
6        packed data instruction.
- 1    3. The machine readable medium Claim 1, when accessed by said processor, further  
2        causing the processor to:  
3            generate the first set of control signals to cause said processor to alter a top of  
4        stack data to indicate that top of the stack of the programmer visible register file is  
5        zero if said first set of bits represents the transition instruction or if said first set of  
6        bits represents said packed data instruction.
- 1    4. The machine readable medium Claim 1 wherein the plurality of control signals stored  
2        thereon are to be accessed by a CISC processor.
- 1    5. The machine readable medium Claim 1 wherein the plurality of control signals stored  
2        thereon are to be accessed by a RISC processor.
- 1    6. The machine readable medium Claim 1 wherein the plurality of control signals stored  
2        thereon are to be accessed by a VLIW processor.
- 1    7. The machine readable medium of Claim 6, when accessed by said processor, further  
2        causing the processor to:  
3            access a second set of bits as representing a floating point state save instruction or

4 a floating point environment save instruction; and  
5 generate a second set of control signals to cause said processor to store into  
6 memory a status data of the programmer visible register file comprising said tag data,  
7 said tag data indicating that the stack of the programmer visible register file is empty  
8 responsive to executing said floating point state save instruction or said floating point  
9 environment save instruction following execution of said transition instruction, and  
10 said tag data indicating that the stack of the programmer visible register file is not  
11 empty responsive to executing said floating point state save instruction or floating  
12 point environment save instruction following execution of said packed data  
13 instruction.

1 8. A computer-implemented method comprising:

2 accessing a first set of bits as representing a packed data instruction;  
3 storing a first corresponding set of control bits to cause a processor to alter a top  
4 of stack data to zero and to operate on packed data contents of a storage representing  
5 a programmer visible register file as a flat register file;  
6 accessing a second set of bits as representing a scalar floating point instruction;  
7 storing a second corresponding set of control bits to cause the processor to  
8 operate on floating point data contents of the storage representing the programmer  
9 visible register file as a stack;  
10 accessing a third set of bits as representing a transition instruction to be executed  
11 between said packed data instruction and said scalar floating point instruction; and  
12 storing a third corresponding set of control bits to cause the processor to alter a  
13 tag data to indicate that the stack of the programmer visible register file is empty.

1 9. The computer-implemented method of Claim 8 wherein the first set of control bits  
2 further causes said processor to alter a sign and exponent data of the programmer  
3 visible register file to indicate not-a-number (NAN).

1 10. The computer-implemented method of Claim 9 wherein said processor is a CISC  
2 processor.

1 11. The computer-implemented method of Claim 9 wherein said processor is a RISC  
2 processor.

1 12. The computer-implemented method of Claim 9 wherein said processor is a VLIW  
2 processor.

1 13. The computer-implemented method of Claim 9 wherein the third set of control bits  
2 further causes said processor to alter the top of stack data to indicate that top of the  
3 stack of the programmer visible register file is zero.

1 14. The computer-implemented method of Claim 9 further comprising:  
2 accessing a fourth set of bits as representing a floating point state save instruction  
3 or a floating point environment save instruction to be executed after said transition  
4 instruction; and  
5 storing a fourth corresponding set of control bits to cause the processor to store a  
6 status data comprising said tag data of the programmer visible register file, wherein  
7 said stored tag data is to indicate that the stack of the programmer visible register file  
8 is empty.

1 15. The computer-implemented method of Claim 9 further comprising:  
2       accessing a fourth set of bits as representing a floating point state save instruction  
3       or a floating point environment save instruction to be executed after said packed data  
4       instruction; and  
5       storing a fourth corresponding set of control bits to cause the processor to store a  
6       status data comprising said tag data of the programmer visible register file, wherein  
7       said stored tag data is to indicate that the stack of the programmer visible register file  
8       is not empty.

1 16. A computer system comprising:  
2       a processor execution unit to execute control signals;  
3       a bus coupled with said processor execution unit configurable to access one or  
4       more machine readable medium;  
5       a first machine readable medium configurable to store a first plurality of  
6       instructions, said first plurality of instructions including a packed data instruction, a  
7       scalar floating point instruction, and a transition instruction to be executed between  
8       said packed data instruction and said scalar floating point instruction;  
9       a second machine readable medium configurable to store a programmer visible  
10      register file and related environment information, said environment information  
11      including a tag data and a top of stack data for accessing said programmer visible  
12      register file in a stack referenced manner; and  
13      a third machine readable medium having stored thereon a plurality of control  
14      signals that, when accessed by the processor execution unit, cause said processor  
15      execution unit to,  
16      access a first set of bits as representing an instruction of the first plurality

17 of instructions from the first machine readable medium,  
18 generate a first set of control signals to cause said processor execution  
19 unit to alter the top of stack data to zero and operate on contents of the programmer  
20 visible register file as a flat register file when said first set of bits represents the  
21 packed data instruction,

22 generate the first set of control signals to cause said processor execution  
23 unit to operate on contents of the programmer visible register file in a stack  
24 referenced manner when said first set of bits represents the scalar floating point  
25 instruction, and

26 generate the first set of control signals to cause said processor execution  
27 unit to alter the tag data to indicate that the stack of the programmer visible register  
28 file is empty when said first set of bits represents the transition instruction.

1 17. The computer system of Claim 16, wherein the plurality of control signals of the  
2 third machine readable medium being accessed by said processor execution unit,  
3 further causes the processor execution unit to:

4 access a second set of bits as representing a floating point state save instruction or  
5 a floating point environment save instruction from the first machine readable  
6 medium, and

7 generate a second set of control signals to cause said processor execution unit to  
8 store said environment information, the tag data indicating that the stack of the  
9 programmer visible register file is empty responsive to executing said floating point  
10 state save instruction or said floating point environment save instruction following  
11 execution of said transition instruction, or otherwise said tag data indicating that the  
12 stack of the programmer visible register file is not empty responsive to executing said

13 floating point state save instruction or floating point environment save instruction  
14 following execution of said packed data instruction.

1 18. The computer system of Claim 16 wherein the plurality of control signals of the  
2 third machine readable medium are to be accessed by a CISC processor execution  
3 unit.

1 19. The computer system of Claim 17 the third machine readable medium comprising  
2 micro code control signals.

1 20. The computer system of Claim 16 wherein the plurality of control signals of the  
2 third machine readable medium are to be accessed by a RISC processor.

1 21. The computer system of Claim 16 the plurality of control signals of the third  
2 machine readable medium comprising VLIW control signals to be accessed by a  
3 VLIW processor.

1 22. The computer system of Claim 20 the third machine readable medium comprising a  
2 read-only memory (ROM).

1 23. The computer system of Claim 21 the first machine readable medium comprising a  
2 dynamic random-access memory (DRAM).